APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: FEATURE-BASED COLOR ADJUSTMENT

APPLICANT: GREGG D. WILENSKY

FEATURE-BASED COLOR ADJUSTMENT

BACKGROUND

The present invention relates to adjusting colors in digital images. Conventional image processing programs provide a number of tools that allow users to adjust colors in

5   digital images. Such color adjustment tools typically operate by changing pixel values (which can represent color, transparency or other information associated with the image pixels) throughout the image, or in a selected region of the image. In conventional operations, colors may be adjusted by changing, for example, the color balance of the image, or the chrominance and/or luminance (or the hue, saturation, transparency, and/or brightness)

10   of colors of selected pixels, or by changing the color values themselves according to a predetermined color adjustment algorithm defined for the particular operation. Particular adjustment operations can include, for example, retouching operations such as smudging, blurring, toning, or sponging, as well as the correction of defects resulting from scratches or dust in selected regions of the image. The amount of the adjustment can be determined based

15   only on information associated with the single pixel to be modified, such as when making a general change in hue, saturation, transparency, luminance, color balance, contrast, for example. Alternatively, the change may be determined by analyzing multiple pixels, such as when applying a Gaussian blur, which determines the amount of change by forming a weighted average value of pixels in the vicinity of the pixel to be changed.

20   Some conventional image processing programs provide tools implementing predefined adjustment algorithms designed to address specific situations or defects commonly found in digital images. One such defect is the redeye effect that can result when the retina of the human eye is exposed to bright light, such as the flash of a camera. This causes portions of the eye, typically portions of the pupil and/or iris, to reflect red light, and

25   therefore appear red in the resulting photographic image. Redeye correction tools typically operate by inspecting individual pixels and/or by applying one or more predefined two-dimensional templates representing various appearances of the human eye to identify a

potential redeye site within a user-specified region of the image, and applying a predetermined correction to the identified redeye pixels.

SUMMARY

In general, in one aspect, the invention provides methods and apparatus, including computer program products, for adjusting the color of pixels in an image. The techniques include identifying a target region of pixels in the image that represent an object, the object having a shape and a predefined set of features; and adjusting the color of one or more pixels in the target region by an amount determined based on one or more probability functions and one or more spatial profile functions, each probability function defining a probability value at each of the one or more pixels in the target region, the probability value representing the probability that the pixel corresponds to one or more features of the object, each spatial profile function being defined based on one or more spatial properties of the object or one or more of its features.

Particular implementations can include one or more of the following features. The spatial properties can include size or shape. The spatial profile function can be a sigmoid function or a Gaussian function. The amount of adjustment can increase as the probability value increases. The amount of adjustment can increase as the probability value decreases. The image can be photographic image including an eye exhibiting a redeye effect; and the identified region of pixels can correspond to a portion of the eye that exhibits the redeye effect. The feature can be one of skin, sclera, or redeye. Adjusting the color can include adjusting the color to match a representative color for the region, and using the probability values to modulate the amount of adjustment. The representative color can represent an iris color for the eye.

In general, in another aspect, the invention provides methods and apparatus, including computer program products, for redeye correction. The techniques include identifying a target region of pixels in an image, the target region corresponding to an eye exhibiting a redeye effect; determining an iris color for the eye; and correcting redeye in the target region by, for every pixel in the target region, blending the color of the pixel with the iris color by

an amount that is based on a redeye probability, the redeye probability representing the probability that the pixel corresponds to redeye.

Particular implementations can include one or more of the following features. Correcting redeye can include avoiding the correction of pixels that correspond to skin or sclera. Correcting redeye can include applying extra correction to pixels that correspond to the ciliary margin of the eye. The method can further comprise eliminating redeye noise that is leftover from the correction.

The invention can be implemented to realize one or more of the following advantages. The use of probability values provides greater control over and allows greater correction of the desired regions while reducing unwanted modifications of the regions that should remain untouched. The use of probability values and spatial profiles increases the accuracy of the color adjustment and provides a smoother, visually more pleasing, result in a larger range of images. The use of probability values and spatial profiles also eliminates the need to identify particular segments of the image. This reduces the amount of computation required for the adjustment and enables adjustment to occur even for images that cannot be segmented.

The details of one or more implementations of the invention are set forth in the accompanying drawings and the description below. Other features and advantages of the invention will become apparent from the description, the drawings, and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of a system for color adjustment in accordance with the invention.

FIG. 2 is a flow diagram of a method of color adjustment in accordance with the invention.

FIG. 3 is a flow diagram of a method of color adjustment in accordance with one implementation of the invention.

FIG. 4 is a flow diagram of a method of color adjustment in accordance with one implementation of the invention.

FIG. 5 is a flow diagram of a method of color adjustment in accordance with one implementation of the invention.

Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

5   FIG. 1 is a block diagram illustrating a functional representation of a color adjustment system 100. The system 100 includes a detection module 110 and an adjustment module 120.

The detection module 110 is operable to receive an image 130. The image 130 can be a raster of rows and columns of pixels. Each pixel has one or more intrinsic properties (e.g.,

10   color, opacity) and each intrinsic property can have one or more associated values. Thus, for example, color can be represented as a set of values in any of a variety of color systems, such as the RGB (red-green-blue) color system, the CMYK (cyan-magenta-yellow-key) color system or the HSV (hue-saturation-value) color system.

The detection module 110 is further operable to scan the image 130 in any orientation

15   (e.g., horizontally or vertically) or direction (e.g., left or right, up or down). The detection module 110 can scan the image pixel-by-pixel, or according to some other pattern (e.g., every other pixel, every other row).

The detection module 110 is further operable to detect an object of interest in the scanned image 130. As used in this specification, an object of interest is an object that can be

20   described by a parameterized set of features that can include primary features that are more indicative of the object and secondary features that are less indicative of the object. The object can also be described in terms of relationships between the features. The relationships can include spatial relationships, for example, proximity between two or more of the features.

The detection module 110 detects the location of the object by identifying pixels in

25   the image that correspond to one or more of the features of the object. A pixel that corresponds to a feature has certain characteristic property values (or ranges of values). For example, where the object is an eye that exhibits a redeye effect, the characteristic red pupil region is a primary feature of the eye and a pixel that corresponds to redeye has the characteristic color value of red.

The output of the detection module 110 can include data that identifies one or more pixels in the image as corresponding to the object. For example, the data can specify the coordinates of the pixel that corresponds to the center of the object. The data can also specify the size of the object. In the case of circular objects, the size can be the radius of the object.

The adjustment module 120 is operable to adjust the color of one or more pixels in the image. One method 200 for performing the adjustment is shown in FIG. 2. The method 200 begins with defining a target region of pixels in the image that represents the object of interest (step 210). The adjustment module 120 can use the output of the detection module 110 to define the target region of pixels. Alternatively, the adjustment module 120 can receive user input defining the target region of pixels.

The method 200 adjusts the color of one or more pixels in the target region based on one or more probability functions (step 220). Each probability function has a value at a given pixel that represents the probability that the pixel corresponds to a particular feature or set of features of the object. For each pixel in the target region or a portion thereof, the method 200 determines an amount by which the color of the pixel will be modified based at least in part on (e.g., as a function of) the probability value or values. In one implementation, the probability values range from 0 to 1, inclusive. The use of probability values enables the color adjustment to be performed on pixels that correspond to particular features while leaving pixels that correspond to other features (or to no particular feature) unadjusted or adjusted by a lesser amount.

In this specification, a probability value for a particular feature will be denoted as "pFeature". A probability value that is denoted as "pFeature1orFeature2" represents fuzzyOr(pFeature1, pFeature2), where fuzzyOr(p1, p2) = p1 + p2 − p1 * p2. Other definitions of the fuzzyOr function can also be used, for example, fuzzyOr(p1, p2) = maximum(p1, p2). The feature can be a feature of the image (e.g., foreground, background) and a corresponding feature probability value can represent the probability that a pixel is (or is not) part of the foreground (or background) of the image. It can also be a feature of an object in the image, for example, where the object is an eye, the feature can be the iris or

sclera of the eye, and a corresponding feature probability value can represent the probability that a given pixel is part of the iris or sclera of an eye.

Different sets of probability values can be used in different portions of the target region. For example, for one portion of the target region, probability values that correspond to a first set of features can be used, while in another portion of the target region, probability values that correspond to a second set of features can be used.

The probability values can be calculated based on one or more probability functions. Alternatively, the probability values can be calculated based on a feed forward neural network. The calculation of probability values is discussed in more detail in co-pending U.S. Application No. ____TBD____, filed on August 11, 2003, entitled "Object Detection Using Dynamic Probability Scans", the disclosure of which is incorporated here by reference in its entirety.

In one implementation, the adjustment module 120 computes a new color value for a given pixel and then blends the new color value with the original color value for the pixel. The new color value can be determined based on a representative color for the region, for example, an average or median color for the region.

The adjustment module 120 uses the probability values to determine the relative contributions of the original and new color values to a resulting color value. For example, the probability values for a pixel can be used as weighting coefficients that determine the relative contribution of each color value (original and new) to the resulting color value for the pixel. The probability values can be used to bias more towards the original color or, alternatively, to bias more towards the new color. These two scenarios are illustrated by the following two functions respectively:

$$c'' = pFeature*c + (1-pFeature)*c'$$

$$c'' = (1-pFeature)*c + pFeature*c'$$

In both scenarios, the resulting color value (c") is the result of combining the original color value (c) with the new color value (c'). In the first scenario, if pFeature is 1, then the original color value is retained and the new color value has no effect on the resulting color value. By contrast, in the second scenario, if pFeature is 1, then the original color value has no effect on the resulting color value and the new color value dominates. If pFeature is less

6

than 1 (but greater than zero), then both the original color value and the new color value have some effect on the resulting color value.

The adjustment amount can also be determined in part based on one or more spatial profile functions (step 230). The spatial profile is a function that has a value at a given pixel, the value being a function of the pixel's location relative to the spatial profile's area of influence. Pixels located beyond the area of influence have a spatial profile value that is zero or close to zero.

More than one spatial profile function can be used, each spatial profile function having a different area of influence. The area of influence for a given spatial profile can be defined based on one or more spatial properties of the object or of its features. The spatial properties include the size and the shape of the object or of its features.

A spatial profile can be implemented, for example, using a pixel mask that is populated with values calculated according to the following function:

Profile = 1-sigmoid(S * (r/R – P)), where sigmoid(x) represents $1/[1 + \exp(-x)]$.

As illustrated by the function, the spatial profile value for a given pixel is a function of the pixel's distance from the center (r), the radius of the target region (R), a steepness value (S) and a proportion value (P). The radial coordinate (r) can be computed as the square root of $[(x – x0)^2 + (y – y0)^2]$, where x0 and y0 are the x and y coordinates of the center of the region of interest, and x and y are coordinates of a pixel in the image.

In the above-illustrated function, the values decrease exponentially as the radial coordinate becomes more distant from the center of the target region. The steepness value determines the steepness of the profile decay relative to the size of the target region. The proportion value determines where the spatial profile value decays to zero. For example, if P=1 then the decay occurs when the radial coordinate is equal to the radius of the target region. If P=2 then the decay occurs when the radial coordinate is twice the radius of the target region.

A spatial profile implementing the above-illustrated function can be used to modulate the adjustment amount such that pixels located farther away from the center of the target region are adjusted by a lesser amount than pixels located closer to the center of the target region. The area of influence can correspond to the target region (or portion thereof) or can

7

extend beyond the target region to produce a smoother, more visually pleasing, effect across the boundaries of the target region.

Other functions can be used to define the spatial profile. Such functions include Gaussian functions, or any function of the pixel coordinates and target region coordinates. Example functions are exponentials, logarithms, sines, cosines, polynomials, and ratios of such functions. Thus, for example, a spatial profile can take the form of a function that decreases monotonically away from the target region center, such as $Profile(r) = exp(-k0\ r)$, or $Profile(r) = 1/(1 + k1\ r + k2\ r*r)$. Similarly, a spatial profile that decreases away from the center, but not monotonically, can be implemented as $Profile(r) = sin(k3\ r)\ exp(-\ k0\ r)$.

Redeye Correction

The above-described techniques can be applied to redeye correction. To compensate for the wide variation of eye colors and shapes, multiple different adjustment steps can be combined to increase the accuracy and completeness of redeye removal across a wide range of images. In one implementation, the redeye correction involves determining an iris color (e.g., blue, brown) and modifying pixel colors in a redeye region based on the iris color. For every pixel in the redeye region, the pixel color is blended with the iris color by an amount that is based on a redeye probability. The higher the probability that the pixel corresponds to redeye, the more influence the iris color has on the resulting color. The modification can be modulated to avoid adjusting (or minimize the adjustment that is applied to) pixels that do not correspond to the iris (e.g., pixels that correspond to the eyelid or to the sclera of the eye). The modulation involves using probability functions to determine the likelihood that the pixel being adjusted corresponds to skin or sclera. The modification can also provide extra correction to specific areas of the eye by using spatial profile functions that are calibrated according to the size and the shape of the specific area. Such areas include, for example, the center of the eye, or the ciliary margin of the eye, which tends to reflect a high concentration of red as a result of the high concentration of muscles in this region. The redeye correction can also use a variant of a median filter to eliminate any specks of red noise that still remain after the above-described modifications have been performed. In one implementation, the correction uses probability values such as those described in the above-referenced patent

application filed on August 11, 2003, entitled "Object Detection Using Dynamic Probability Scans".

As shown in FIG. 3, a method 300 in accordance with this implementation of the invention begins when the adjustment module 120 receives input that identifies a region of

5 redeye to be corrected (step 310). This region will be referred to as a redeye circle. In one implementation, this input is received from the detection module 110 and the input identifies the center pixel location and the radius, R, of the redeye circle.

The adjustment module 120 defines a circular region of pixels in the image that corresponds to the iris of the eye (step 320). This region will be referred to as an iris circle.

10 The iris circle is centered on the redeye circle and extends beyond the redeye circle by a pre-determined amount. For example, the iris circle can be defined as 1.5 times the size of the redeye circle. Defining the iris circle to extend beyond the redeye circle improves the completeness of the correction and also improves the appearance of the correction by smoothing the effect of the correction across the edges of the redeye circle.

15 The adjustment module 120 iterates through the pixels within the iris circle and adjusts the colors for each pixel in the window (step 330). As shown in FIG. 4, the first step of the adjustment involves converting the color values for each pixel from RGB color space values of (red, green, blue) to perceptual lab color space values of (l, a, b) (step 410). The value "l" is a luminance value and "a" and "b" are chrominance values. The following

20 conversion formulas can be used:

$$l = (1/sqrt(3)) * \log[(L+1)(M+1)(S+1)];$$
$$a = (1/sqrt(6)) * \log[(L+1)(M+1)/(S+1)^2]; \text{ and}$$
$$b = (1/sqrt(2)) * \log[(L+1)/(M+1)].$$

L, M, and S represent retinal cone response values and serve as intermediate values in the

25 conversion process. The values for L, M, and S can be determined as follows:

$$L = 0.3811 * red + 0.5783 * green + 0.0402 * blue;$$
$$M = 0.1967 * red + 0.7244 * green + 0.0782 * blue; \text{ and}$$
$$S = 0.0241 * red + 0.1288 * green + 0.8444 * blue.$$

The conversion to a perceptual color space enables chrominance to be adjusted separately

30 from luminance.

The adjustment module defines an annulus at the rim of the iris circle (step 420). In one implementation, the annulus is offset from the redeye circle by a distance of one pixel, so the inner radius of the annulus is equal to R + 1. The outer radius of the annulus is set to maximum[minRimThickness, (rimExpansionFactor-1.0) * (R+1)], where the

5   minRimThickness is equal to 2 and the rimExpansionFactor is equal to 1.1. This enables the outer radius to scale in proportion to the size of the redeye circle. Setting rimExpansionFactor equal to 1.1 causes the annulus to have a thickness of approximately 1/10 the radius of the redeye circle.

The adjustment module 120 determines representative chrominance values, <aRim>,

10   <bRim>, within the annulus (step 430). The representative chrominance values represent the iris color of the eye. The representative chrominance values can be determined by calculating a weighted average <a> and <b> of each of the chrominance values:

<a>= Sum[a weight]/Sum[weight]; and

<b> = Sum[b weight]/Sum[weight].

15   The summation is performed over all pixels in the annulus. The weight equals 0 if pSkinOrRedeye > 0.5 and the weight equals saturation * (1 − pSkinOrRedeye) otherwise. Saturation can be set to abs(a) + abs(b), where abs(x) represents the absolute value of the variable x. The use of the probability values pRedeye and pSkin enables the adjustment module 120 to skip pixels that correspond to redeye or skin.

20   Alternatively, the representative chrominance values can be determined using other formulas for calculating means or median values (e.g., by taking the square root of the sum of the squares).

The adjustment module 120 uses the representative chrominance values to calculate an adjusted color for each pixel in the iris circle (step 440). The adjusted color is initialized

25   to the chrominance values and then modulated according to one or more probability values and spatial profiles.

For example, as illustrated in the following function, the pRedeye probability value can be used to leave non-red pixels unadjusted:

a' = pRedeye * aRim + (1.0 − pRedeye) * a; and

30   b' = pRedeye * bRim + (1.0 − pRedeye) * b.

In these functions, a' and b' are the adjusted color values whereas a, and b are the original color values.

One difficulty encountered in redeye correction is that portions of the eyelid or of the sclera (white of the eye) may overlap with the redeye circle. To avoid adjusting pixels that correspond to skin or sclera, the adjustment module 120 uses the probability values pSkin and pSclera to further modulate the adjustment as illustrated by the following functions:

$$a'' = pSkinOrSclera * a + (1 - pSkinOrSclera) * a'; and$$
$$b'' = pSkinOrSclera * b + (1 - pSkinOrSclera) * b'.$$

In these functions, a and b are the original chrominance values, a' and b' are the initial adjusted values (after the redeye adjustment) and a'' and b'' are the final adjusted values (after the skin and sclera adjustment).

An annular region of pixels at the perimeter of the redeye circle generally corresponds to the ciliary margin region of an eye. This region often contains redeye. To correct the redeye in this region, the adjustment module 120 desaturates the color of the pixels in this region.

In performing the desaturation, the adjustment module 120 uses a spatial profile, ciliaryProfile, which has a pre-determined area of influence that corresponds to the dimensions of the ciliary margin region. The dimensions can be defined empirically, for example, based on measuring the ciliary margin for one or more sample eyes. In one implementation, the ciliaryProfile is defined as exp(-kCiliary * (r/R - 1.0)*(r/R - 1.0)). This produces a profile which takes on its maximum value of 1 when the radial coordinate, r, is equal to the iris radius, R. The profile takes on a value of less than 1 when r differs from R. The value of kCiliary determines the strength of the decay. Setting the value of kCiliary to $0.5/(0.2)^2$ causes the profile to have a specified thickness of roughly 20% of R.

In performing the desaturation, the adjustment module 120 also uses the probability values pSkin and pHighlight to avoid adjusting skin or highlight pixels, as illustrated below:

$$a''' = a'' * [pSkinOrHighlight + (1 - pSkinOrHighlight) * (1 - ciliaryProfile)].$$
$$b''' = b'' * [pSkinOrHighlight + (1 - pSkinOrHighlight) * (1 - ciliaryProfile)].$$

The center of the redeye circle often has the highest concentration of the red pixels. To correct the redeye in this region, the adjustment module 120 can use another spatial profile, centerDesaturationProfile, that has an area of influence corresponding to this region. The centerDesaturationProfile can be defined as sigmoid(10.0 * (r/R – 0.3)). These values provide significant desaturation to roughly 30% of the iris radius and produce fairly steep falloff which gives a visually pleasing result.

The redeye effect often causes the typically dark pupil to become overly bright. To correct the redeye in this region, the adjustment module 120 uses another spatial profile, pupilProfile, which is defined as sigmoid(5.0 * (r/R – 0.5). The adjustment module 120 also uses the probability value of pHighlight to avoid darkening highlights, as illustrated below:

l' = l * [pHighlight + (1.0 - pHighlight) * (darkenFactor + (1.0 – darkenFactor) * pupilProfile)]. Setting the darkenFactor to .3 causes the adjusted luminance (l') to be 30% of the original luminance (l) in the middle of the pupil.

Once the adjusted color values for each pixel have been computed, the adjustment module 120 converts the adjusted color values from the perceptual color space back to the RGB color space (450). The following conversion formulas can be used:

red =   4.46867 * L - 3.58868 * M + 0.119604 * S;

green = -1.21972* L + 2.383088 * M - 0.16263 * S; and

blue = 0.058508 * L - 0.26108 * M + 1.205666 * S.

L, M, S can be defined as follows:

L = exp[(2 l' sqrt(3) + a' sqrt(6) + 3 b' sqrt(2))/6.0] - 1;

M = exp[logL – b' sqrt(2)] - 1; and

S = exp[l' - logL – logM] – 1.

In these functions, l' is equal to l*sqrt3, a' is equal to a*sqrt6 and b' is equal to b*sqrt2.

The adjustment module 120 then adjusts the color values of each pixel in the iris circle to match the adjusted color values (step 460). In performing the adjustment, the adjustment module 120 applies a spatial profile, circleProfile, that has an area of influence corresponding to the entire iris circle:

red' = circleProfile * red + (1 – circleProfile);

green' = circleProfile * green + (1 – circleProfile); and

blue' = circleProfile * blue + (1 − circleProfile).

CircleProfile can be defined as 1 - sigmoid(5.0*(r/R − 1.5)). This profile function is an example of a more general profile function that has a value of 1 at the circle center, r = 0, and decays towards 0 at a larger radius, here chosen to be 1.5*R. The steepness of the profile decay has been set to 1/5 of the circle radius, R. This profile function smoothes the transition between the region within the iris circle and the outer regions of the image which remain unadjusted.

To improve the completeness of the redeye correction, the adjustment module 120 can perform an additional adjustment step to remove more redeye. This additional adjustment is illustrated by the method 500 of FIG. 5.

For each pixel that lies within the iris circle, the adjustment module 120 centers an evaluation window on the pixel (step 510). The size of the window can be scaled in proportion to the size of the iris circle. For example, the window dimensions can be defined as follows: The windowDiameter can be set to 2 * windowRadius + 1; and the windowRadius can be set to 1 + int(irisCircleRadius)/10. This sets the window size to be 1/10 the size of the irisCircle and also constrains the window radius to be no smaller than 2 and no larger than 5. If the window size is too small, then accuracy is compromised, but if the window size is too large, then speed is compromised.

Within the window, the adjustment module 120 gathers a list of chrominance values for each chrominance value a and b (step 520). The list of values can be represented in the form of two arrays: aw[i] and bw[i]; where i represents the array element number, which can range from 0 to nWindowPoints - 1, where nWindowPoints = windowDiameter^2.

The adjustment module 120 then selects a representative chrominance value from each list (step 530). The representative chrominance value can represent the median or mean color value for the pixels within the window. However, the presence of redeye within the window will cause the median or mean color to be more red than normal. Thus, to offset the redness from the redeye, a weighted median or mean value can be selected by first sorting each list and then choosing the list element at list position nWindowPoints/4. This biases more towards choosing less red colors.

To avoid adjusting highlights, the adjustment module can use the probability value pHighlight to adjust the color of the representative color values (step 540).

a' = pHighlight * a + (1 - pHighlight) * awSorted[index]; and

b' = pHighlight * b + (1 - pHighlight) * bwSorted[index].

5   The adjustment module 120 then adjusts the color values of the pixel to match the representative color values (step 550). In performing the adjustment, the adjustment module 120 applies a spatial profile, circleProfile that has an area of influence corresponding to 1.33*R: circleProfile = 1.0 - sigmoid(5.0 * (r/R − 1.33)). This adjustment is illustrated below:

10   red'' = red' * circleProfile + red * (1 - circleProfile);

green'' = green' * circleProfile + green * (1 - circleProfile); and

blue'' = blue' * circleProfile + blue * (1 - circleProfile).

Brush implementation

15   The above-described adjustment techniques can be implemented as a multi-profile brush tool that can apply multiple brush profiles during a single invocation of the brush tool. Each brush profile defines the shape of the brush's region of influence and the adjustments that are applied. The shapes and adjustments can differ from profile to profile.

Each of the above-described color adjustments can be implemented as a separate 20 profile that is applied by the multi-profile brush. For example, the application of a spatial profile can be implemented as a Gaussian brush profile which produces a stronger effect at the center of the brush and reduced effect towards the edge of the brush. The location of the region of interest is determined by the brush center coordinate.

The invention can be implemented in digital electronic circuitry, or in computer 25 hardware, firmware, software, or in combinations of them. Apparatus of the invention can be implemented in a computer program product tangibly embodied in a machine-readable storage device for execution by a programmable processor; and method steps of the invention can be performed by a programmable processor executing a program of instructions to perform functions of the invention by operating on input data and generating output. The 30 invention can be implemented advantageously in one or more computer programs that are

executable on a programmable system including at least one programmable processor coupled to receive data and instructions from, and to transmit data and instructions to, a data storage system, at least one input device, and at least one output device. Each computer program can be implemented in a high-level procedural or object-oriented programming

5 language, or in assembly or machine language if desired; and in any case, the language can be a compiled or interpreted language. Suitable processors include, by way of example, both general and special purpose microprocessors. Generally, a processor will receive instructions and data from a read-only memory and/or a random access memory. The essential elements of a computer are a processor for executing instructions and a memory. Generally, a

10 computer will include one or more mass storage devices for storing data files; such devices include magnetic disks, such as internal hard disks and removable disks; magneto-optical disks; and optical disks. Storage devices suitable for tangibly embodying computer program instructions and data include all forms of non-volatile memory, including by way of example semiconductor memory devices, such as EPROM, EEPROM, and flash memory devices;

15 magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROM disks. Any of the foregoing can be supplemented by, or incorporated in, ASICs (application-specific integrated circuits).

To provide for interaction with a user, the invention can be implemented on a computer system having a display device such as a monitor or LCD screen for displaying

20 information to the user and a keyboard and a pointing device such as a mouse or a trackball by which the user can provide input to the computer system. The computer system can be programmed to provide a graphical user interface through which computer programs interact with users.

The invention has been described in terms of particular embodiments. Other

25 embodiments are within the scope of the following claims. For example, steps of the invention can be performed in a different order and still achieve desirable results. The invention can also be used to change eye color, for example, to replace the iris color with a user-specified color. It can be used to modify lip color or lipstick color on lips. It can also be used to remove or reduce defects on skin given a suitable measure of skin defect

30 probability. It can be used to modify (e.g., brighten, saturate, recolor) the color of flowers

15

given a measure of probability which distinguishes flower petals from non-petals. What is claimed is: